Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 46 (2015) 732 – 741
Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

The Interplay between Requirements Relationships Knowledge and Requirements Change towards Software Project Success: An Assessment Using Partial Least Square (PLS)

Ruhaya Ab. Aziz[a,*], Bernard Wong[b]

[a]UTHM, Lock Bag 101,Parit Raja, Batu Pahat, 86400, Malaysia/ UTS, Sydney
[b]Enterprise Strategy Consulting, PO Box 628, Wahroonga, NSW 2076, Australia

Abstract

Changing of requirements in a software project should be managed effectively. However, managing changes is a challenge due to many reasons. One of the reasons is requirements do not stand alone and they are typically related to one another in several ways. The relationships may impact individual requirement as well as the entire software project. Thus, this research aims to investigate how these types of requirements relationships impact requirements change as well as software project success. We examined the impacts from the perspective of business analyst using PLS. The findings can be used as a guide on working with requirements relationships knowledge that is useful for business analysts and research community.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Requirements Relationships Knowledge; Requirements Change; Success

1. Introduction

Success is a major concern for any stakeholders in a software development project. Accordingly, there are various factors contribute to project success discussed in software engineering literature. One of the many factors is

* Corresponding author. Tel.: +6-07-4533765; fax: +6-07-4536023.
E-mail address: ruhaya@uthm.edu.my

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.138

the degree of requirements change[1]. Ebert and De Man[2] state that “a key reason for project failures is insufficient management of changing requirements during all stages of the project lifecycle”. The later researchers indicate effective change management as one of the 26 critical success factors[3]. In contrast, there are other researchers who argue that impacts of requirements change are not always significant in ensuring software development project success as long as the requirements are considered complete at some point during the project[4]. Software is malleable and changes will always possible during software development project. Changes in software are always asserted as related to the continuous changes in requirements[5]. Once software is deployed into use, new requirements will emerge and existing requirements may change. Changes in business, technology and stakeholder needs will always generate new requirements for existing software [6-7]. Moreover, design, implementation and testing activities provide additional feedbacks which also contribute to the changes in requirements [6]. Upon these changes, part of the software may have to be altered to correct errors that are discovered in operation, to adapt it for a new platform and to improve performance [7]. Hence, software development is a continuous process which will not stop although a software system is delivered. These incremental changes are critical to be managed to ensure that the software will remain useful.

Managing these incremental changes in software development is very challenging. The knowledge and management of requirements changes is crucial in the management of software changes [8]. However, the knowledge of requirements change is not enough where we need to know how the requirements related to one another. Furthermore, as requirements change, there is a need to understand what happens to existing relationships between those changing requirements. Consequently, these relationships may change and these changes have to be traced. In this paper, the information of the relationships between requirements is addressed as requirements relationships knowledge (RRK). RRK is concerned with how requirements are related to one another and other artefacts during the software development project. Although RRK is not problematic, the knowledge would affect other aspects of software development project and the project as a whole [9-14].

There are studies that focus on how requirements change impacts success [1, 4, 15]. Other studies focus on how RRK impacts requirements change [9, 11, 12]. However there are limited studies that focus on how RRK will impact requirements change as well as the success of software development project. Thus, this paper will continue to investigate further into this issue. The rest of this paper is organised as follows: Firstly, The research context and research model will be presented in Section 2. Secondly, research method, which mainly concerns on the development and the validation of the requirements relationships instrumentation design, will be presented in Section 3. Finally, Section 4 will present a discussion and conclusion including the implications of the work both in research and practice.

2. Research Model

Recently, the interrelationships between RRK and requirements change has been addressed further by other researchers. Most of the researchers have focused on how to utilize the RRK for change propagation analysis and extended the investigation for change impact analysis. Checik et al[16] use model based approach and provide an automated algorithm for changes propagation between requirements and design model. In the study they have used activity diagram and sequence diagram to enable the automated change propagation. Moreover, Zhang et al[17] also utilized RRK for change propagation analysis. They used a case study to evaluate the applicability of two well known generic dependencies modeling covering about 25 dependencies types and then propose new classification model. Another body of knowledge defines taxonomy of requirements relationships use in application to identify traces between requirements elements to architecture elements [18]. They have used the granularity of the dependency taxonomy as predictor to determine architectural elements that are more receptive to change [18]. Ali, Rozan and Sharif [19] then use the interrelationships between artefacts as the basis for change impacts analysis. They then propose process of impact analysis based on literature and interview with project leaders and related stakeholders. RRK is also used as the basis for change propagation analysis in requirements [20]. In the study, they use Sysml to represent textual requirements and then used design structure matrix to trace the change propagation in the requirements specification. Those recent literatures had shown the significant of RRK in requirements change management in which seem that most of them focusing on the use of RRK in change propagation analysis and moreover change impact analysis. Thus, the related hypothesis is:

H1: RRK has significant impacts on requirements change

Furthermore, a software development project can be defined as a set of activities inclusively both technical and managerial, require in satisfying the terms and conditions of a project agreement in developing a software within budget and schedule [21]. Accordingly, success is typically defined based on how the software development project meeting the established budget, time and requirements. However, there are also other definitions introduced in literature. The other literature defined success as quality of product [22-23], satisfaction of the stakeholder [22] and scope [24]. On the other hand, Linberg [25] argues that a project can only be regarded as a success when the product meets the quality expectation whereas a cancelled project can only be regarded as a failure when there is no learning could be applied to the next project. Although, there are many definitions and criteria proposed to measure success of a software development project, the criteria will always involve requirements. Indeed, Leffingwell and Widrig [26] indicate that the most frequent and serious problems associated with software development are related to requirements. The requirements initially defined will always change and this will impact cost, schedule and other factor of success [1]. Thus, it is important to manage requirements change to reduce the ripple effect. The related hypothesis will be:

H2: Requirements Change has significant impacts on success

Accordingly, as RRK has significant impact on requirements change and requirements change has significant impacts on the success of software development project, this paper intends to investigate further and hypothesize that RRK will indirectly has significant impacts on the success of software development project (H3). The relationship is represented as dotted line in the research model as shown in Fig. 1. Furthermore, this paper also hypothesizes that requirements change mediate the relationships between RRK and the success of software development project.

H3: RRK has significant impacts on requirements change and success of software development project
H4: Requirements Change will mediate the relationships between RRK and software development project success

The overview of the research model and the related hypotheses are illustrated in Fig. 1. The constructs and the items represent the construct in the research model are further illustrated in Fig. 2. This paper will continue with the discussion on how the research model is validated and tested especially about the research method used in the next section.

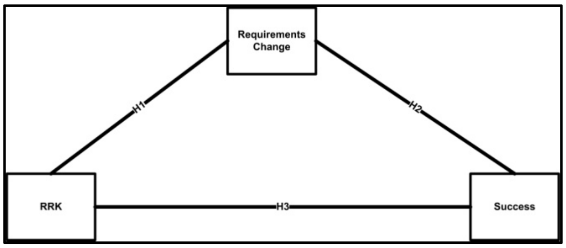

Fig. 1. The proposed model of RRK

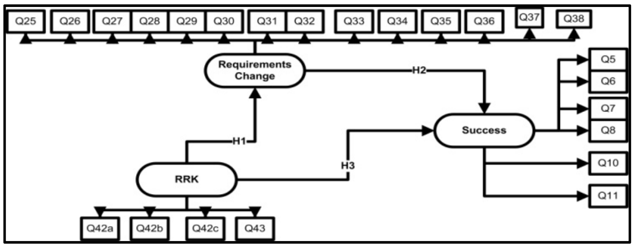

Fig. 2. Part of RRK Model and the initial constructs

3. Research Method

3.1. Participants

In this study, non-probability sampling of purposive sampling was used. Business analysts and any stakeholders involved in requirements in their software development project were chosen and those not involved were excluded from the sample. 173 business analysts and related stakeholders (to the software development projects) were recruited. Inclusion criteria were as follows: the participants were stakeholders involved in requirements management in their software development project. 60% of the participants were business analysts and system analysts (refer to Table. 3.). About 10-20% of the respondents are from Australia and most of them are practitioners in Malaysia industry. The survey data show that, most of the respondents are from the medium and large organisation (refer to Table. 1. and Australian Bureau of statistics classification of business framework (office of small business, 1999). Accordingly, in Table.2., the survey data shows that the industry domain of most of the organization is from Information Technology and Telecommunication (41%) and Infrastructure and government (30%). In addition, the respondents mostly have about 2-5 years (37%) and about 6-9 years (25%) experience in writing requirements which represent about 62% of all respondents (Table. 4.).

Table 1.Employment Information (source: Survey data, 2011-2012)

Items	Frequency	Percentage
Less than 10	15	9%
11-100	84	49%
101-500	32	18%
More than 500	41	24%

Table 2.Industry domain (source: Survey data, 2011-2012)

Items	Frequency	Percentage
Factory automation	3	1.7=2
Financial	10	5.7=6
Infrastructure & Government	52	30
Medical	4	2.3=2
IT & Telecommunication	72	41.1=41
Transportation	2	1
Others	32	18.3=18

Table 3.Designation of the respondents (source: Survey data, 2011-2012)

Items	Frequency	Percentage
Business Analyst	18	10.5
Business Analyst Manager	6	3.5
System Analyst	80	47
System Designer	20	11.6
Tester	4	2.3
Others	43	25.1

Table 4.Experience in Writing Requirements (source: Survey data, 2011-2012)

Items	Frequency	Percentage
Within one year	32	19
2-5 years	63	37
6-9 years	43	25
10-15 years	23	13
More than 15 years	8	5

3.2. Structural Equation Modeling and Partial Least Square (PLS)

This study is a part of a research that examined the impacts of requirements relationships on the other elements and activities in a software development project that may impact success. The activities and elements may have direct and indirect relationships; they will impact each other and thus the success of a particular software development project as a whole. Therefore, in order to validate and investigate the relationships further and the impacts that they give to one another, Structural Equation Modelling (SEM) was used. SEM is a statistical technique for testing a theoretical model hypothesized by a researcher using a combination of statistical data and qualitative causal assumption [27]. This approach is more confirmatory rather than exploratory, thus is more suitable for theory testing than theory development. SEM is a very general, powerful multivariate analysis technique that comprises specialised versions of a number of other analysis methods as special cases [27]. SEM is not designed for a single statistical technique but it is also a family of related procedures [27].

Furthermore, SEM can be categorised into two approaches, which are: 1) covariance-based approach, which is related to tools such as EQS and Analysis of Moment Structures (AMOS); and 2) variance-based approach, which is related to PLS. Thus, in this research, PLS was chosen. In this research, Partial Least Square (PLS) was used because of the following reasons: 1) research on requirements relationships is relatively new; and 2) there is no measurement model that is already available. PLS could be a suitable technique to be used when the phenomenon to be investigated is relatively new [28]. Thus, the assessment of the goodness of measure of these constructs in terms of their validity and reliability within the research framework will be presented in the next section.

3.3. Measures and Goodness of Measures

A questionnaire using five-point Likert scale was used to collect data for each construct of the research model. Some of the instruments were newly developed where most of the questions were created based on the theory from literature and other empirical studies. There were also some instruments that were adapted from previous literature. The final constructs of the model are illustrated in Table. 5.

Table 5.Part of the constructs in the Model

Construct	Item	Description
Success	Q5 (SC1)	The outcome of the project meets the business goal.
	Q6 (SC2)	The outcome of the project meets all the specified requirements.
	Q7 (SC3)	The overall quality of the developed application / product is high.
	Q10 (SC4)	The project is completed within scope.
	Q11 (SC5)	The requirements-related tasks (e.g. requirements specification, requirements management) have been completed successfully in the project.
Requirements Change	Q38 (RC3)	Every change in requirements is documented so that it is easy to track what change has been made.
	Q29 (RC1)	Any changes in requirements slow down the schedule of the project.
	Q30 (RC2)	Any changes in requirements will increase the cost spent in the project.
Requirements Relationships Knowledge	Q42a (RRK1)	The relationships between requirements that exist between the components are considered when deciding to implement the solution.
	Q42b (RRK2)	The relationships between requirements that exist between the components are considered when planning the schedule for the design/development team to complete the task.
	Q35 (RRK3)	Before implementing a change to a particular requirement, any possible impact it will cause to other requirements will be considered.

Accordingly, two main criteria are used for evaluating goodness of the measures, which are validity and reliability. The combination of both is essential to ensure the quality of a research [29]. The validity and reliability measures of this research model are discussed in the next section.

3.4. Construct Validity

Construct validity is concerned with the degree to which interferences can legitimately be made from the operational constructs in a study to the theoretical constructs on which those operational constructs are based on [29]. Sekaran and Bougie [30] indicate that construct validity is used to testify how well the results obtained from the use of the measure fit the theories around which the test is designed. Thus, to assess how the instrument fits the concept as theorised, convergent and discriminant validity can be used. Firstly, the respective value of loadings and cross loadings in Table. 6. were examined to assess whether there were any problems with any particular items. A cut-off value for loadings at 0.5 was considered as significant [31]. If there were any items with a loading of higher than 0.5 on two or more factors, then they were deemed to be having significant cross loadings. From Table.6., we can see that all the items that measured a particular construct would load highly on the construct and would have lower loadings values on other constructs hence confirming construct validity.

Table 6. Loading and Cross Loading

	RC	RRK	SC
RC1	0.683	0.194	0.175
RC2	0.623	0.096	0.136
RC3	0.845	0.307	0.373
RR1	0.278	0.926	0.246
RR2	0.300	0.930	0.247
SC1	0.269	0.249	0.716
SC2	0.321	0.181	0.802
SC3	0.287	0.279	0.782
SC4	0.146	0.159	0.690
SC5	0.306	0.116	0.733

3.5. Convergent Validity

Secondly, the validity test was continued with the convergent validity. It is the degree to which multiple items are in agreement to one another in measuring the same concept. Factor loadings, composite reliability, and average variance extracted (AVE) were used to assess the convergent validity. This practice was proposed by Hair et al. [31]. In this research, the test showed that the factor loadings for all items exceeded the recommended value of 0.5 [31]. Next, composite reliability values (refer to Table. 7.), which illustrate the degree to which the construct indicators indicated the latent, ranged from 0.764 to 0.925. This exceeds the recommended value of 0.7 [31]. Finally, the average variance extracted measured the variance captured by the indicators relative to measurement error. It should be greater than 0.5 to justify the use of the construct [32]. As shown in Table. 7., the AVE was in the range of 0.523 to 0.861.

Table 7.Results of measurement model

Model Construct	Measurement item	Loading	CR[a]	AVE[b]
Requirements Change (RC)	RC1	0.683	0.764	0.523
	RC2	0.623		
	RC3	0.845		
Requirements Relationships Knowledge (RRK)	RRK1	0.926	0.925	0.861
	RRK2	0.930		
Success	SC1	0.716	0.862	0.556
	SC2	0.802		
	SC3	0.782		
	SC4	0.690		
	SC5	0.733		

Table 8.Summary Results of the Model Construct

Model Construct	**Measurement Item**	**Standardised Estimate**	**T-Value**
Requirements Change	RC1	0.683	5.963
	RC2	0.623	6.819
	RC3	0.845	2.831
Requirements Relationships Knowledge	RRK1	0.926	38.286
	RRK2	0.930	42.992
Success	SC1	0.716	13.616
	SC2	0.802	22.313
	SC3	0.782	19.591
	SC4	0.690	11.708
	SC5	0.733	14.106

Moreover, the results for the measurement model are summarised in Table. 8.. The results show that all of the 3 constructs: Requirements Change, Requirements Relationships Knowledge and Success were all valid measures of their respective constructs based on their parameter estimates and statistical significance.

3.6. Discriminant Validity

Thirdly, the test was continued to validate the discriminant validity. Discriminant validity is the degree to which items differentiate among constructs where they show the measures that theoretically should not be related are in reality not related. This validity test was assessed by examining the correlations between measures of potentially overlapping constructs. The items should have the highest loading value on their own constructs in the model, and the average variance shared between every construct and its measures should be greater than the variance shared between the construct and other constructs [33]. Table.9. shows that the squared correlation for each construct is less than the average variance extracted by the indicators measuring the construct indicating adequate discriminant validity. As a result, the measurement model demonstrates adequate convergent validity and discriminant validity.

Table 9.Discriminant Validity of Constructs

	1	2	3
1. Req. Change	**0.523**		
2.Req. Relationships	0.097	**0.861**	
3. Success	0.130	0.070	**0.556**

3.7. Reliability Analysis

Reliability is about the quality of measurement. Reliability in a research is the extent to which a measurement procedure yields the same answer however and whenever it is carried out [34]. Reliability is about the quality of measurement. Reliability in a research is the extent to which a measurement procedure yields the same answer however and whenever it is carried out [34]. One of the general classes of reliability is the internal consistency reliability that is used to assess the consistency of result across items within a test [29]. In this research, Cronbach's alpha coefficient was used to assess the reliability of the inter item consistency of the measurement items. Table.10. shows the summarisation of loadings and alpha values. All the alpha values listed in Table. 10. are above 0.6, which are conforming to what have been suggested by Nunnaly and Berstein [35]. The composite reliability values also ranged from 0.764-0.925. Composite reliability is another approach similar to Cronbach's alpha for the estimation of internal consistency reliability. In the approach, a composite reliability value of 0.7 or more is considered acceptable [36]. Thus, it can be concluded that the measurements were reliable.

Table 10.Results of Reliability Test

Constructs	Measurement items	Cronbach's Alpha	Loading range	Num. of items
Req. Change	RC1, RC2, RC3	0.618	**0.643-0.922**	3(6)
Req. Relationships	RRK1, RRK2	0.839	**0.926-0.930**	2(3)
Success	SC1, SC2, SC3, SC4, SC5	0.799	**0.690-0.820**	5(7)

3.8. Hypothesis Testing and Mediation Effect Analysis

The result from the analysis shows that there are significant relationships between requirements relationships knowledge, requirements change and success of software development project. Fig.3. illustrates the analysis which shows that the initial coefficient for the three constructs. The analysis shows that the path coefficient value for RRK->RC is 0.312 and the path coefficient for RC-> success is 0.314. Both values are more than the range of (0.20-0.30) in which has been indicated as acceptable [28]. Thus, it can be concluded that there are significant relationships exist between the three constructs in which supporting the three hypotheses as stated in section 2.

In addition, mediator effect analysis has also been conducted. The analysis reports that, there exists a mediator relationship between the three constructs. In order to allow for mediator analysis, there are certain criteria that need to be fulfilled. Firstly, the predictor (RRK) has significant impact on the mediator requirements change (RC);

secondly, the mediator (RC) has significant impact on the criterion variable success; third, the predictor (RRK) has significant impact on the criterion variable in the absence of the mediators' impact. Thus, to establish the mediating effect, the indirect effect of a x b (see Fig. 3.) has to be significant. In this regard, the z statistic is applied [37], in which the value is significant at $p < 0.05$. If the z value exceeds 1.96 ($p < 0.05$), then the hypothesis 1 and 2 can be accepted where there is an indirect impact of RRK through requirements change on success of software development project. The z value is defined as equation (1):

$$z = \frac{a \times b}{\sqrt{b^2 \times s_a^2 + a^2 \times s_b^2 + s_a^2 \times s_b^2}} \tag{1}$$

As shown in Fig. 3., there is a significant impact of RRK on requirements change (0.312, $p<0.05$) as well as requirements change on success (0. 314, $p<0.05$). Due to the reason that, there is also a significant direct impact of RRK on success of software development project (0.172, $P<0.05$), requirements change is established as a partial mediator. This mediating effect of requirements change is confirmed by z statistic [37] as shown in equation (2):

$$z = \frac{0.312 \times 0.314}{\sqrt{0.314^2 \times 0.068^2 + 0.312^2 \times 0.084^2 + 0.068^2 \times 0.084^2}} = 2.898 \tag{2}$$

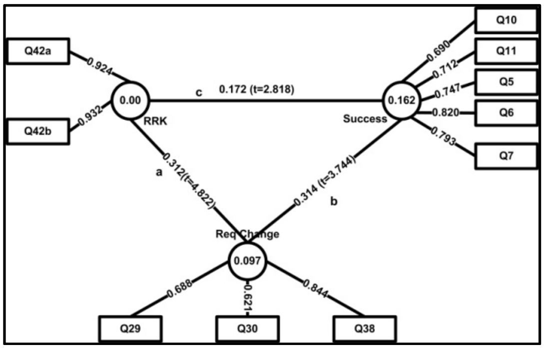

Fig. 3. Requirements Change as mediator

The result shows that, requirements change has mediating effects in which it implies that there is indirect impact on success. Accordingly, variance accounted (VAF) value is used to represents the ratio of the indirect effect to the total effect. The VAF value indicates that 36.3% of the total effect of RRK on success of software development project is explained by indirect effect (requirements change):

$$\frac{\mathrm{a} \times \mathrm{b}}{a \times b + c} = \frac{0.312 \times 0.314}{0.312 \times 0.314 + 0.172} = 0.363 \tag{3}$$

Thus, it can be concluded that the relationships between the three constructs is significant and confirmed by the mediation effects that exist among them.

4. Discussion and Conclusions

This study emphasizes on the impacts of the independent variables of RRK on requirements change and the success of software development project using the PLS technique. It is a part of a research that examines the impacts of RRK on several of software development success factors. In this paper, the goodness of measure is also assessed by looking at the validity and reliability of the measures using the PLS approach. The results show that the measures used demonstrate both convergent validity and discriminant validity. Moreover, the reliability of the measures was also assessed by looking at the Cronbach Alpha values and Composite Reliability values. As a result, both the Chronbach Alpha values and Composite Reliability values have fulfilled the criteria set up by other established researchers. The results have shown that the measures in the model were reliable.

The findings from the analysis indicate that a software development project is considered successful when the project has achieved several criteria, which are: 1) the outcome of the project meets all the specified requirements; 2) the overall quality of the product is high; 3) The requirements-related tasks (e.g. requirements specification, requirements management) are successfully completed in the project; 4) The outcome of the project meets the business goal; and 5) The project is completed within scope. All the criteria are in fact quite similar to the criteria suggested by previous researchers (e.g. [23, 24]). The findings indicated that, as long as the outcome meets all the specified requirements and business goals, has good quality, completes within scope, and all the requirements-related activities are completed successfully, the project will be considered successful although the project is not completed within time and budget. Thus, the findings apparently suggest that the success of requirements activities in which including managing requirements change will impact success.

Moreover, the findings of the paper confirmed views that RRK has significant impacts on requirements change. The results of the analysis also confirmed the hypothesis that highlight the importance of RRK as one of the significant predictors for success of software development project. Results of the analysis confirmed the direct impact of RRK has on requirements change is significant. Firstly, RRK has significant impact on requirements change in which inline with what has been indicated in the literature [16-20]. The knowledge of how requirements related to one another provide guide on how a set of requirements can be organized and structured in requirements document. The good structure and organization of requirements can contribute to the good quality of requirements [9, 14]. Consequently, it will help to track any changes and documents the changes accordingly. According to the analysis of the result, the main characteristics of requirements change that related to RRK are: 1) Every change in requirements was documented so that it is easy to track what change has been made; 2) Any changes in requirements may slow down the schedule of the project 3) any changes in requirements will add to the cost spent in the project. The first characteristic is actually focused on the interrelationships between RRK and requirements change where the way we organize and document the requirements and every change in the requirements may help requirements change management. The second and third characteristics are more focused on how requirements change may impacts success.

In conclusion, the findings confirmed the phenomena of the interrelationships between RRK->Requirements change ->Success. The findings also confirmed the four (4) hypotheses listed in this study i.e. requirements relationships knowledge has indirect significant impact on the success of a software development project. As requirements relationships knowledge has significant impact on requirements change (H1), and requirements change has direct significant impacts on success (H2) and mediate the interrelationships between RRK and success (H4), it can be concluded that requirements relationships knowledge is another significant predictor that will impact requirements change as well as project success (H3). In future, this study continues to examine further this quantitatively finding with a qualitative study in investigating how RRK impacts requirements change and other related factors to achieve success from the perspectives of business analyst.

Acknowledgements

The author is grateful to the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Tun Hussein Onn, Malaysia for providing the facilities to carry out the research.

References

1. Gorschek T, Davis AM. Requirements Engineering: In search of the dependent variable. *Information and Software Technology* 2008; **50**: 1-2. p. 67-75.

2. Ebert C, De Man J. Requirements Uncertainty: Influencing factors and Concrete Improvements. *Proceeding of 27th International Conference on Software Engineering*, Missouri: IEEE; 2005.
3. Nasir MHD, Sahibuddin S. Critical success factors for software project: A comparative study. *Scientific Research and Essays* 2011; **6**:10. p. 2174-2186.
4. Verner J, Cox K, Bleistein S, Cerpa N. Requirements Engineering and Software Project Success: An Industrial Survey in Australia and the U.S. *Australasian Journal of information systems* 2005; **13**: 1.p. 225-238.
5. Rajlich VT. Changing the Paradigm of Software Engineering. *Communication of the ACM* 2000; **49**:8. p. 67-70.
6. Anderson S, Felici M. Requirements Evolution from Process to Product Oriented Management. In: Bomarius F, Komi-Sirvi S. Editors. *Product Focused Software Process Improvement*. Berlin: Springer-Verlag; 2001: **2188**. p. 27-41.
7. Sommervile I. Software Engineering 9th Edition. London: Pearson Education Limited; 2011.
8. Lam W, Shankararaman V. Requirements Change: A Dissection of Management Issues. *Euromicro Conference Proceedings* 1999; **2**: p.2244-2251.
9. Karlsson L, Dahlstedt A, Regnell B, Dag JNO, Persson A. Requirements Engineering challenges in market driven software development-An interview study with practitioners. *Elsevier Science Direct - Information and Software Technology* 2007; **49**: 2007. p. 588-604.
10. Svensson RB, Olsson T, Regnell B. An Investigation of how quality requirements are specified in industrial practice. *Information Software Technology* 2013; **55**: 7. p. 1224-1236.
11. Davis AM. Just Enough Requirements Management: Where software development meets Marketing. New York: Dorset House Publishing; 2005.
12. Dahlstedt AG, Persson A. Requirements Interdependencies: State of The Art and Future Challenges. In Aurum A, Wohlin C. Editors, *Engineering and Managing Software Requirements*. Germany: Springer Berlin Heidelberg; 2005. p. 95-116.
13. Ozkaya I. Representing Requirements Relationship. *First International Workshop on Requirements Engineering Visualisation*. IEEE Computer Society 2006; p.3.
14. Diev S. Structuring complex requirements. *ACM SIGSOFT Software Engineering Notes 2007;* **32**:2 p. 1-5.
15. Singh MP, Vyas R. Requirements Volatility in software development process. *International Journal of soft computing & Engineering* 2012; **2**: 4. p. 259-264.
16. Ali HA, Abd Rozan M, Sharif AM. Identifying Challenges of Change Impact Analysis for Software Project. *Proceeding of International Conference on Innovation, Management and Technology Research*. Malaca Malaysia: IEEE; 2012.
17. Checik M, Lai W, Nejati S, Cabot J, Diskin Z, Easterbrook S, Sabetzadeh M, Salay R. Relationship-Based Change Propagation: A Case Study. *Workshop on Modelling in Software Engineering*. ICSE 2009. Vancouver. Canada. IEEE. 2009.
18. Zhang H, Li J, Zhu L, Jeffery R, Liu Y, Wang Q, Li M. Investigating dependencies in Software Requirements for Change Propagation Analysis. *Information and Software Technology* 2014; **56**:1.p.40-53.
19. Khan SS, Lock S. Concern Tracing and Change Impact Analysis: An Exploratory Study. *Proceeding of workshop on Aspect Oriented Requirements Engineering*. ICSE 2009. Vancouver. Canada: IEEE; 2009.
20. Nonsiri S, Coatance E, Bakhouya M, Mokammel F. Model-Based Approach for Change Propagation Analysis in Requirements. Proceedings of IEEE International System Conference. Orlando FL: IEEE; 2013.
21. IEEE Standard Software Project recommended practices.
22. Thomas G, Fernandez W. Success in IT projects: A matter of definition?. *International Journal of Project Management 2008*; **26**: 7. p.733-742.
23. Agarwal N, Rathod U. Defining success for software projects: An exploratory revelation. *International Journal of project management 2006;* **24**: p. 358-370.
24. Wohlin C, Mayrhauser AV, Höst M, Regnell B. Subjective evaluation as a tool for learning from software project success. *Information & Software Technology* 2000; **42**: 14. p. 983-992.
25. Linberg KR. Software Developer Perceptions about Software Project Failure: a case study. *Journal of Systems and Software* 1999; **49**: p.177-192.
26. Leffingwell D, Widrig D. *Managing Software Requirements: Second Edition (A use case approach)*. Boston: Pearson Education; 2003.
27. Kline RB. *Principles and Practice of Structural Equation Modeling*. New York: The Guilford Press; 1998.
28. Chin WW. Issues and Opinion on Structural Equation Modelling. *MIS Quarterly 1998*; **22**:1. p. 7-16.
29. Trochim WMK, Donnelly JP. *The Research Methods knowledge Base*, Third edn. USA: Atomic Dog a part of CENGAGE Learning; 2008.
30. Sekaran U, Bougie R. *Research Methods for Business: A Skill Building Approach*. UK: John Wiley and Sons; 2010.
31. Hair JF, Black WC, Babin BJ, Anderson RE. *Multivariate Data Analysis*. Upper Saddle River. NJ: Prentice-Hall; 2010.
32. Barclay DW, Thompson R, Higgins CA. The Partial Least Squares (PLS) Approach to Causal Modeling: Personal Computer Adoption and Use an Illustration. *Tech Studies* 1995; **2**:2. p. 285-309.
33. Compeau DR, Higgins CA, Huff S. Social Cognitive Theory and individual Reactions to Computing Technology - A Longitudinal-Study. *MIS Quarterly* 1999; **23**:2. p.145-158.
34. Kirk J, Miller ML. *Reliability and Validity in qualitative research*: *Series 1*. USA: Sage Publications; 1986.
35. Nunnally J, Berstein I. *Psychometric Theory*. New York: McGraw-Hill; 1994.
36. Fornell C, Larcker DF. Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research* 1981; **18**:1.p. 39-50.
37. Preacher KJ, Leonardelli GJ. 2013. at http://www.quantopsy.org/sobel/sobel.htm access on 13 February 2013.